

<研究ノート>

N88 日本語 BASIC (MS—DOS 版) による
データ処理

BASICによるテスト処理(3)

難 波 俊 裕

入学直後に一年生全員に課せる統一テストの処理は、C. S. 社が、昭和 63 年 4 月まで数年間、担当した。この採点処理は、回答の分析と共に、ハード・コピーの形で納品される。しかし、われわれの目指すものは、一時的評価ばかりでなく、少なくとも一般教養課程 2 年間の継続的学力の追跡であるため、それをコンピュータで処理可能なデータとすることが求められる。処理を担当する C. S. 社は、処理データも企業秘密であると称して、われわれの要請に対して、処理完了のものでなく、未処理のデータのはいったフロッピー・ディスクを辛うじて、提供してくれるにすぎない。

提供されるデータは、処理されていないばかりでなく IBM コード (EBCDIC: Expanded Binary-Coded Decimal) で記録されているため、OS である MS-DOS の ASCII (American National Standard Code for Information Interchange) に変換しなければならない。最初の頃、一度 N 88 (86) DISK-BASIC の ASCII データに変換し、それを更に MS-DOS の ASCII に再変換するという煩雑な手続きを必要した。昨年入手した『よこぐるま』では、EBCDIC を MS-DOS のコードに直接変換が可能であり、この作業を軽減してくれた。

これを改めて処理したものは、最終的にはランダム・ファイルに格納されるのであるが、処理の過程と使用するプログラムを紹介するのが本稿の

目的である。

I MS-DOS と BASIC. COM

MS-DOS は、もともと Micro-Soft 社が開発し、IBM 社がパソコン用に PC-DOS として 1983 年に採用して以来、急速に広まった。NEC が MS-DOS を PC98 系のパソコンに採用したのも、殆ど同じ頃である。1985 年、NEC は、この DOS の上で機能する BASIC COMPILER を発売した。これは次の組合せで、P コードの Object File を生成するものであって、

N88BASIC. COM

N88BASIC. LIB

BASIC. COM

この三組の Version が一致しないと、コンパイルド・モードでは、正確に作動しないことは勿論であるが、Vr. 1 は PC98VM ではカラー・モードでプログラムしたにも拘らず、発色しなかった。Vr. 2 は、一見健全そうではあったが、PC9801 の上では計算機能が十分発揮できず、ストップした。Vr. 4 が出現するまでは、プログラムが完成しても機種によって、改めて機種に適した Version でコンパイルを仕直す手間が避けられなかった。P コードの実行プログラムは、コンパイルされていると言っても、依然、解釈のために参照される N88 BASIC. LIB がカレント・ドライブに存在しないと作動しない。

基本的な上述の障害に加えて、N88 BASIC. COM と N88 BASIC. LIB との上に構築される BASIC プログラムと EXECUTION プログラムとの間には、文法の相違もある。その一は、同時 OPEN のファイル数の指定とランダム・ファイルを使用するときのレコード・サイズを指定する方法の相違である。BASIC モードでは、コマンド N88BASIC を入力するとき、2 個のスイッチ [/F: ファイル数 /S: レコードサイズ] の書式で指定するが、一方、EXECUTION モードでは、REM 文で [\$FILE ファイル数 (,) レコード・サイズ] と記入しておかなければならない。

また、デバッグ、プログラム修正の時、BASIC モードと MS-DOS モー

ドとを頻繁に交替しなければならないが、その対策にチャイルド・プロセスが一応は設けられてはいる。しかし、これよりも、バッチ・ファイルの方が便利である。例えば [N88 BASIC /F: 4 /S: 128] を組み入れた BASIC. BAT を作っておくと、MS-DOS で EXECUTION プログラムを走らせた直後に、簡単に BASIC モードに移行できるからである。

もう一つの煩雑は、DOUBLE PRECISION などの変数の型式の指定の相違である。[DEFINT / SNG / DBL / STR] の型宣言が BASIC モードでは使用できるのに反して、コンパイルド・モードでは型宣言文字 (% , ! , # , \$) による指定に限って有効であるにすぎない。

これらの制約にもかかわらず、コンパイルされた EXECUTION ファイルの利点は、高速処理が実現するだけでなく、数種のプログラムを組み合わせ、バッチ・ファイルを作り、コマンド入力の手間をはぶいて一連の処理ができることである。統一テスト(英語)のように、学部ごとに同じ処理手順を繰り返す場合に BATCH FILE によるプログラムの差替えは、大変効率がよい。

II YOKOGURUMA

「よこぐるま」は、4種のファイル、つまり、EBCDIC, N88DISK-BASIC, MS-DOS, CP/M86 のデータを、そのいずれにも、変換できる FILE CONVERTER である。

FILE の構造には2種あって、STREAM 型には変換方式を特定しなくてよいが、FIX 型には特定仕様のカタログと呼ぶ型を予め設定して、変換する。われわれの作業には、漢字系のコードはないので、カタログの作成の必要がない。STREAM 型とは、レコードの構造が256バイト以下で、末尾に Cursor Return (DH) Line Feed (AH) を持つ、シーケンシアル・ファイルである。

入力ドライブと出力ドライブは自由に設定できるので、8インチ・ディスクと5インチ・ディスクとの間の交換にも便利である。ただし、「よこぐるま」自体は、大小文字の相違を区別する N 88 (86) DISK-BASIC を OS

としているから、ファイル名を大文字で入力しておかないと、MS-DOS ファイル名が小文字となってしまう、データを引き出せないことになり、次の作業に支障が生じる。出力、入力データとそのドライブ、ボリューム単位かファイル単位か、ファイル形式、コード変換の有無、出力ファイル名を指示すれば、ファイル変換が実施される。

その際のコンソールの表示は次のようになる。

データ・コンバージョン	EBCDIC	→	MS-DOS
	(ドライブ3)		(ドライブ2)
機能	:	ファイル単位	
ファイル形式	:	Stream	
コード変換	:	有り	
カタログ名	:	未使用	
入力ファイル名	:	DATA	
出力ファイル名	:	UNITST88	

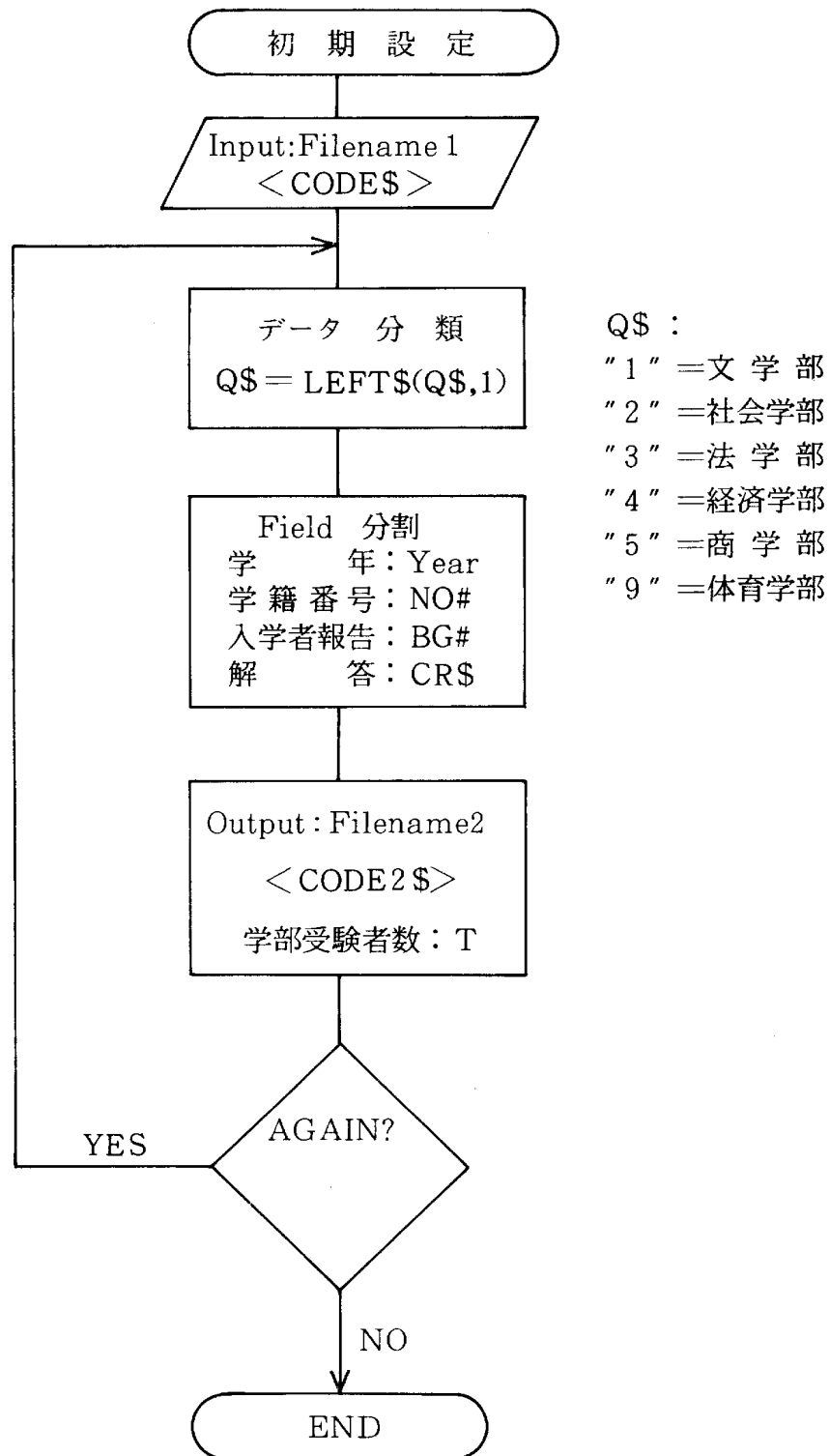
Ⅲ FILREAD

EBCDIC コードから ASCII コードに変換した DATA は1レコード80バイトからなるシーケンシャル・ファイルであるが、これを学年、学籍番号、入学者報告、回答の項目 (Fields) に分解し、学籍番号を手掛りに、更に、学部別のファイルに編成するプログラムである。レコードは学年が1バイト、学籍番号、7バイト、入学者報告6バイト、回答が40バイト、合計は、実質54バイトのデータである。

プログラム・ディスクをAドライブ、データ・ディスクをBドライブに設置し、読み取るべき資料ファイル名 (CODE\$) と受験者総数を入力、ついで、編成する学部ファイル名 (CODE2\$) を入力すれば、資料レコードを上記の4項目に分割しメモリーに格納し (※ BREAK: 850—1110)、最後に、これをシーケンシャル・ファイルとしてBドライブのディスクに記入する (OUTPUT AS#2: 740—790)。

そのフロー・チャートは次の通りである。

FILREAD



```

100 'save "filread",a
110 REM-----READ IBM FILE ONLY   File Name "FILREAD" V.3.0-----
120 '
130 '   List of Variables
140 '
150 '   YEAR(n)      :Year Grade i.e. Y$
160 '   NO$(n)       :Presence Number of Students i.e. NO$
170 '   BG$(n)       :School Background of Students i.e. BG$
180 '   CR$(n)       :Untreated Answers of a Student
190 '   T            :Total of Treated Students
200 '   A$(n)        :Tips of Dat$, of which Y$,NO$,BG$ & CR$ are made.
210 '   Q$           :The 2nd Tip of Dat$, meaning the Dept, or 1st of Pres. No.
220 '   DAT$         :IBM Data of a Student
230 '   CODE$        :File Name of Data converted from IBM code
240 '   CODE2$       :File Name of Departmental Data
250 '   By this program, the sequential Data file will be separated
260 '   into Departmental Files with Student No, Background, CDR Data of
265 '   40 answers.
270 '
280 CONSOLE 0,25,0,1,1
290 REM $file 2,128
300 CLS
310 PRINT "*****"
320 PRINT "This program is just to Read a File put from IBM CODE into MS-DOS"
330 PRINT "*****"
340 INPUT "Are sure to run this ? (Y/N)";Z$:IF Z$="Y" OR Z$="y" THEN ELSE 830
350 COLOR 6
360 FILES "b:"
370 COLOR 7
380 INPUT "What is the Data File Name to be read...";CODE$
390 *AGAIN
400 PRINT "Which Dept are you going to take in and make a file of?"
410 PRINT "1) Bungaku Bu   2) Hougaku Bu   3) Shougaku Bu"
420 PRINT "4) kezai gaku Bu   5) Taiikugaku Bu"
430 PRINT "6) Shakaigaku Bu"
440 INPUT "...Input the number";AA
450 IF AA=1 THEN CODE2$="bun"
460 IF AA=2 THEN CODE2$="hou"
470 IF AA=3 THEN CODE2$="sho"
480 IF AA=4 THEN CODE2$="kez"
490 IF AA=5 THEN CODE2$="tai"
500 IF AA=6 THEN CODE2$="sha"
510 DIM YEAR(900),NO$(900),BG$(900),CR$(900)
520 OPEN "B:"+CODE$ FOR INPUT AS #1
530 FOR I=1 TO 3000
540 INPUT #1,DAT$
550 PRINT DAT$
555 Q$=MID$(DAT$,2,2)
560 '
570 IF AA=1 THEN Q$=LEFT$(Q$,1):GOTO 630
580 IF AA=2 THEN Q$=LEFT$(Q$,1):GOTO 640
590 IF AA=3 THEN Q$=LEFT$(Q$,1):GOTO 650
600 IF AA=4 THEN Q$=LEFT$(Q$,1):GOTO 670
610 IF AA=5 THEN Q$=LEFT$(Q$,1):GOTO 660
620 IF AA=6 THEN Q$=LEFT$(Q$,1):GOTO 680
630 IF NOT(Q$="1") THEN 710 ELSE 690
640 IF NOT(Q$="3") THEN 710 ELSE 690
650 IF NOT(Q$="5") THEN 710 ELSE 690
660 IF NOT(Q$="9") THEN 710 ELSE 690
670 IF NOT(Q$="4") THEN 710 ELSE 690
680 IF NOT (Q$="2") THEN 710
690 T=T+1
700 GOSUB *BREAK
710 IF EOF(1) THEN 730
720 NEXT I
730 CLOSE #1
740 OPEN "B:"+CODE2$ FOR OUTPUT AS #2
750 WRITE #2, T
760 FOR I=1 TO T
770 WRITE #2, YEAR(I),NO$(I),BG$(I),CR$(I)
780 NEXT I
790 CLOSE #2
800 PRINT "Are you going to make a file of another dept?"
810 INPUT "..... (Y/N)";AA$
820 IF AA$="Y" OR AA$="y" THEN T=0:ERASE YEAR,NO$, BG$,CR$:GOTO *AGAIN
830 '
840 END
850 *BREAK
860 DIM A$(60)
870 FOR J=1 TO 54
880 A$(J)=MID$(DAT$,J,1)
890 NEXT J
900 Y$=A$(1):YEAR(T)=VAL(Y$)
910 NO$=""
920 FOR H=2 TO 8

```

```
930 NO$=NO$+A$(H)
940 NEXT H
950 NO$(T)=VAL(NO$)
960 BG$=""
970 FOR G=9 TO 14
980   BG$=BG$+A$(G)
990 NEXT G
1000 BG$(T)=VAL(BG$)
1010 CR$=""
1020 FOR F=15 TO 54
1030 IF A$(F)="9" THEN A$(F)="-"
1040   CR$=CR$+A$(F)
1050 NEXT F
1060 CR$(T)=CR$
1070 COLOR 4
1080 PRINT YEAR(T);NO$(T);BG$(T);CR$(T),T
1090 COLOR 7
1100   ERASE A$
1110 RETURN
```

IV TRETAPR

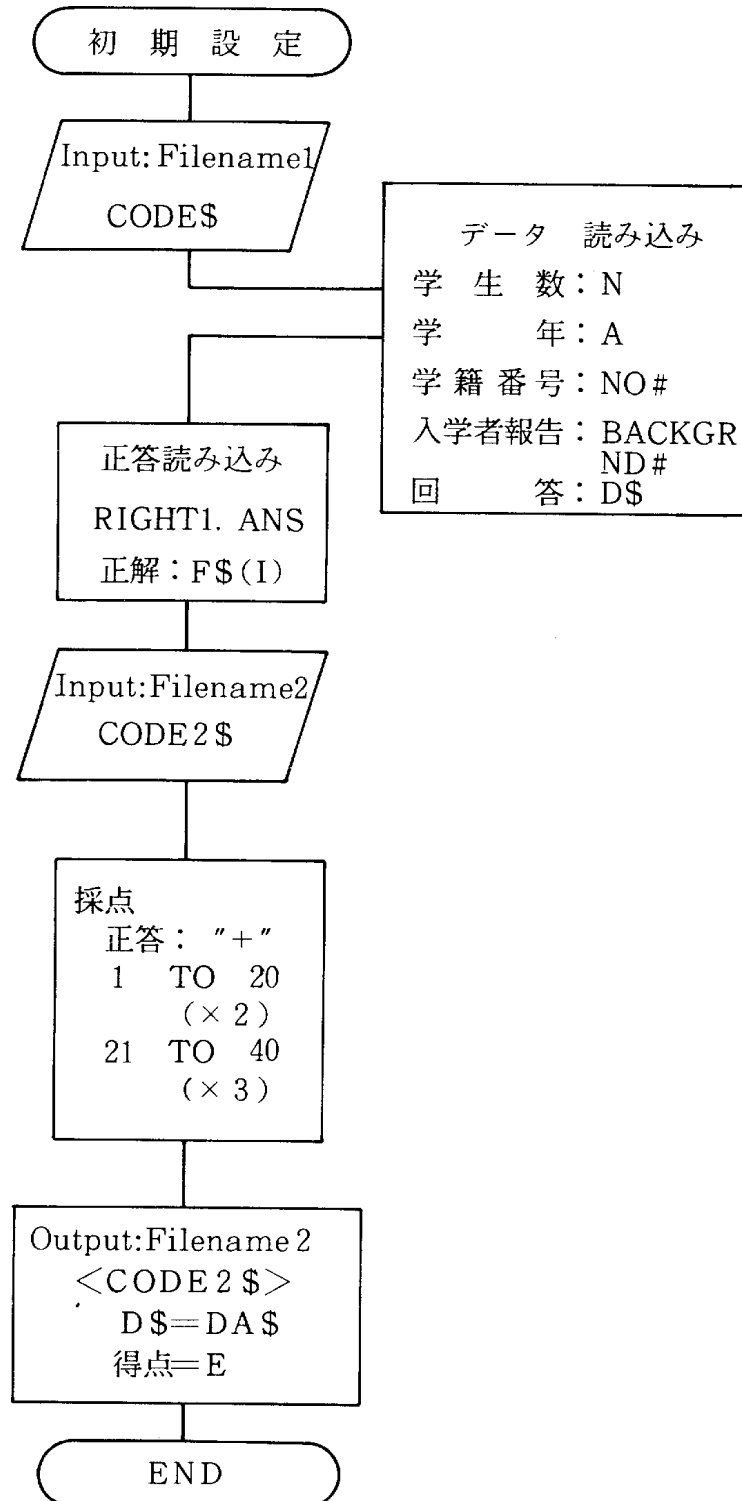
前章の FILREAD で作成した学部各のファイルを読み取り、採点結果を別のシーケンシャル・ファイルとして、B ドライブに記入するプログラムである。

読み取るべきファイル名 (CODE\$) を入力し、次で、A ドライブの正解を読み込み、40 問について、各正答 F\$ (I) と回答 DF\$ (I) を照合する。その際、問 1 から問 20 までは配点 2 点、問 21 から問 40 までは配点 4 点として得点を合計する (790—900)。同時に、正解であれば '+' 誤答であれば、誤答の選択番号をそのまま残しておく文字列 DA\$ を作り、後の SP 分析の資料として、新しいファイル名で B ドライブに記入する。

これは、4 月の統一テスト専用の採点プログラムである。

フローチャートは次の通り。

TRETAPR




```

100 'save"tretapr",a
110 REM -----Mark & Treat Seq. Data File into Marked File -----
120 '       File Name: "TRETAPR", V. 3.00
130 '       List of Variables
140 '       CODE$ :The code of the file to be treated
150 '       A(n)  :Year Grade i.e. YEARS$
160 '       NO#(n) :Attendance No.
170 '       C(n)  :Background of Students i.e. BG$
180 '       D$(n) :Series of Answers of the Student i.e. CR$
190 '       F$(i) :Fragments of Right Answer
200 '       DF$(i) :Data Fragments of Student Answers to be checked by F$
210 '       FF$   :New String of Right Answer
220 '       DD$(j) :Already Checked Fragments of Student, to make DA$
230 '       DA$   :Will be made into New 'D$(n)'
240 '       T     :Points Acquired by a Student
250 '       E(n)  :T
260 '       F(n)  :Value (T*3/10) of a Student
270 '
280 CLEAR
290 REM $file 2,128
300 '-----FILE READER-----
310 CONSOLE 0,25,0,1
320 '
330 PRINT "You are to Mark & Treat Dept Data file into New Seq. File"
340 PRINT "Are you Ready?"
350 INPUT "                               Yes(1)/No(0)";RY:IF RY<>1 THEN 990
360 COLOR 6
370 FILES "b:*.*)"
380 '
390 COLOR 7
400 PRINT"Which Dept Data do you like to treat?"
410 INPUT "Input the File Name          ";CODE$
420 CLS
430 LOCATE 0,0,1
440 COLOR 6:PRINT USING"          Codes:&          &          ";CODE$
450 COLOR 7 :CONSOLE 2,23,0,1
460 LOCATE 0,3,1
470 DIM A(900),NO$(900),BACKGRND$(900),D$(900),E(900),F(900)
480 '
490 GOSUB 1050
500 '
510 '
520 '
530 '----- CALL RIGHT ANSWERS-----
540 '
550 '
560 DIM F$(50)
570 OPEN "A:RIGHT1.ANS" AS #1
580 FIELD #1,4 AS RC$,4 AS Y$,8 AS NO$,50 AS CR$
590 GET #1,1
600 NO.RIGHT$=NO$
610 PRINT NO.RIGHT$,CR$
620 PRINT "... Hit 'Retrn Key' if you are sure.":LINE INPUT A$
630 FF$=""
640 FOR I=2 TO 41
650 F$(I)=MID$(CR$,I,1)
660 FF$=FF$+F$(I)
670 NEXT I
680 CLOSE #1
690 '-----TREATED FILE (Sequential Access) -----
700 INPUT " Input the 2nd File Name, ex(Ta187.Apr)";CODE2$
710 PRINT USING" CODE2$ :&          &";CODE2$
720 PRINT "          Sure? Hit 'Retrn' if yes.": A$=INPUT$(1)
730 IF A$=CHR$(13) THEN ELSE 700
740 OPEN "B:"+CODE2$ FOR OUTPUT AS #1
750 WRITE #1,N
760 FOR I=1 TO N
765 PRINT I
770 DIM DF$(50),DD$(50)
780 DA$=""
790 FOR J=1 TO 40
800 DF$(J)=MID$(D$(I),J,1)
810 IF DF$(J)="9" THEN DF$(J)="-"
820 IF DF$(J)=F$(J+1) THEN DD$(J)="-" ELSE DD$(J)=DF$(J)
830 DA$=DA$+DD$(J)
840 NEXT J
850 FOR J=1 TO 20
860 IF DD$(J)="-" THEN T=T+2
870 NEXT J
880 FOR J=21 TO 40
890 IF DD$(J)="-" THEN T=T+3
900 NEXT J
910 D$(I)=DA$: E(I)=T :F(I)=T*.3
920 GOSUB 1170

```

```

930 PRINT "-----"
940   ERASE DF$,DD$:T=0
950 NEXT I
960   CLOSE #1
970   '
980   '
990   '
1000 END
1010   '
1020   '
1030   '
1040   ' ----- (For April File ) -----
1050   '
1060 OPEN "B:"+CODE$ FOR INPUT AS #1
1070   INPUT #1,N
1080 FOR I=1 TO N
1090   INPUT #1,A(I),NO#(I),BACKGRND#(I),D$(I)
1100   '
1110   PRINT " ";NO#(I);" ";D$(I);BACKGRND#(I)
1120 NEXT I
1130   CLOSE #1
1140   RETURN
1150   '
1160   '
1170   PRINT NO#(I);BACKGRND#(I);D$(I);E(I);F(I)
1180   WRITE #1,NO#(I),BACKGRND#(I),D$(I),E(I),F(I)
1190 RETURN
1200   '
1210   '
1220   '
1230   '

```

V TRANSFIL

後述する NUMBR2 というプログラムで、予め作成しておく DATA ファイルに、採点後の学部別のファイルのデータを、所定のフィールドに格納するプログラムである。

DATA ファイルは学籍名簿を参照して予め正確なものにしておくので、仮に学籍番号の誤記のデータがあれば、それは DATA ファイルに転写されない。そのデータがあるか、ないかも、検証するサブルーチンを具えている (※ UNFINISH: 1250—1310)。

取り込むファイル名は拡張子が“APR”と“JAN”の二種類あり、CODE\$はこの区別を入力しなければならない。また、格納する Field も、統一テスト (E1\$), 一年生後期共通テスト (E2\$), 二年生後期共通テスト (E3\$) を区別する必要がある (510—560)。

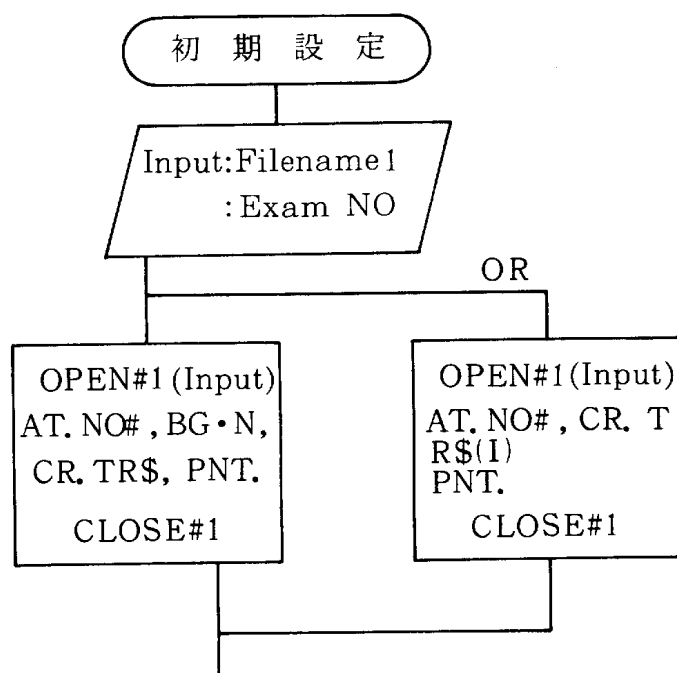
DATA ファイルはランダム・ファイルの構造を有するデータ・ベースで、拡張子 `DAT` で区別される。その項目 (Field) の配分は次の通りである。

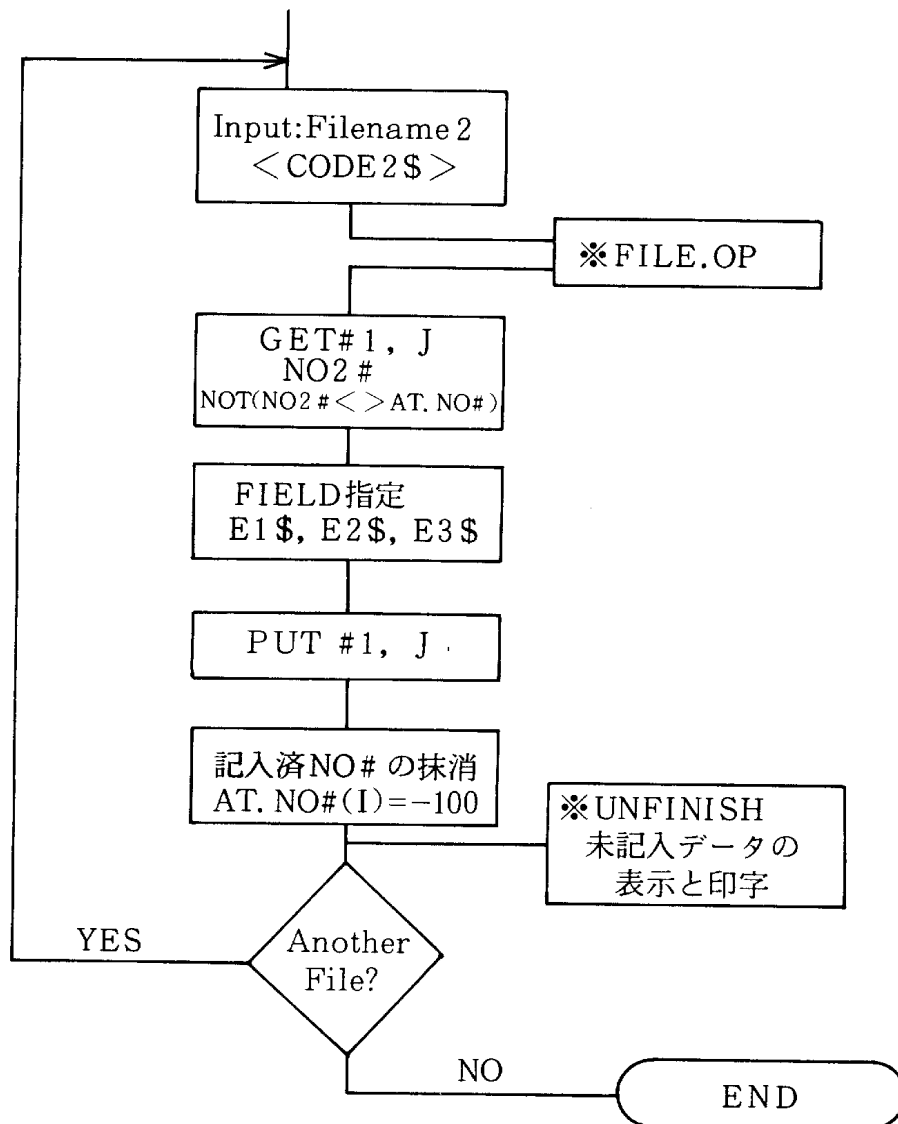
FIELD #1, 4 AS RC\$, 8 AS NO\$, 12 AS SEI\$, 12 AS MEI\$, 2 AS CL1\$, 2 AS CL2\$, 4 AS E1\$, 4 AS E2\$, 4 AS E3\$, 4 AS E4\$, 4 AS E5\$, 8 AS BG\$, 42 AS CdR\$, 4 AS SUM\$, 4 AS AVE\$, 4 AS ORD\$, 4 AS OPT\$

DATA ファイルの学籍番号 NO2# が読み込みファイルの学籍番号 AT.NO# と一致すれば, DATA ファイルに記入される (PUT #1, J: 930), と同時に, この AT.NO# は抹消されるから, 抹消されずに残った AT.NO# があるかどうかを検証サブルーチン (※ UNFINISH) で検証された場合には, 1) 該当学部受験カードを調べて正しい学籍番号を調査するか, 2) 不受験として処置された学生の申告を待つか, の対策しかない。1) の手段は相当手間を要する作業となることを覚悟しなければならない。しかし, 重複とちがって, このケースは非常に少ないので, 2) の方法も, 妥当である。

フローチャートは次の通りである。

TRANSFIL





```

110 REM -----To read treated Data & transmit it to Source File (Apr & Jan)
120 '      File Name: "TRANSFIL", V. 3.00
130 '      List of Variables
140 '      CODE$      :The code of the file to be treated
150 '      YR(n)      :Year Grade i.e. YEAR$
160 '      AT.NO(n)   :Attendance No. i.e. NO, NO$
170 '      BG.N(n)    :Background of Students i.e. BG$
180 '      CR.TR(n)   :Series of Answers of the Student i.e. CR$
190 '      PNT(N)     :Points Acquired by a Student
200 '      CODE2$     :The code of the Random File for data to be described in
210 '      RC2        :Record No of the Source file
220 '      NO2        :Attendance No of the Source file
230 '      RC$        :R.File Descriptor of Record No
240 '      NO$        :R.File Descriptor of Attendance No
250 '      E1$        :R.File Descriptor of Exam 1 i.d. one taken in April
260 '      BG$        :R.File Descriptor of School Background
270 '      CDR$       :R.File Descriptor of Series of Marked Answers,CR$
280 '
290 '
300 CLEAR
310 '-----FILE READER-----
320 '
330 CONSOLE 0,25,0,1,1
340 REM $file ,128
350 LOCATE 0,0
360 PRINT "Program: Transmit Files"
370 COLOR 4

```

```

380 PRINT "To Trasmit the treated Data from S.Q.file (*.APR or .JAN) to Source F
ile(*.DAT)"
390 COLOR 7
400 PRINT "Are you Ready?"
410 PRINT
420 INPUT "                                Yes(1)/No(0)";RY:IF RY<>1 THEN 1100
430 CONSOLE 2,23,,1
440 CLS
450 PRINT"Which Dept Data do you like to transmit?"
460 COLOR 6
470 FILES "B:*.apr"
480 FILES "B:*.jan"
490 COLOR 7: PRINT TAB(20);:INPUT "Input the filename. Ex. 'BUN.APR(JAN)':" ;CODE
$
500
510 PRINT "Which Exam is this ?"
520 PRINT " 1) No.1 exam.          2) No.2 exam.          3) No.3 exam."
530 INPUT "          Choose the number";EXAM
540 PRINT TAB(10);USING" .... CODE$ &          & EXAM NO ##";CODE$;EXAM
550 PRINT TAB(20);:INPUT"... Are you sure? (Y / N)";A$
560 IF A$="Y" OR A$="y" THEN ELSE 440
570 DIM RC(900),AT.NO#(900),BG.N(900),CR.TR$(900),PNT(900)
580 LOCATE 0,0 : COLOR 6
590 PRINT USING" Program : Transfil.Exe(Bas)---DATA taken from &          & : Exa
m: #";CODE$;EXAM : PRINT
600 CONSOLE 2,23,0,1,0 : COLOR 7
610 N=0
620 IF EXAM=2 OR EXAM=3 THEN GOTO 1330
630 OPEN "b:"+CODE$ FOR INPUT AS #1
640 INPUT #1,N
650 FOR I=1 TO N
660 INPUT #1, AT.NO#(I),BG.N(I),CR.TR$(I),PNT(I),THRD
670 PRINT " ";AT.NO#(I);BG.N(I);CR.TR$(I);PNT(I)
680 NEXT I
690 CLOSE #1
700 '-----Transmitting to Source file (*.DAT)-----
710 '
720 PRINT "Which file are you going to ?"
730 COLOR 2
740 FILES "b:*.dat"
750 COLOR 6
760 PRINT TAB(20):INPUT " Key in your filename(ex. LAW86.DAT).";CODE2$
770 '
780 COLOR 7
790 PRINT TAB(25): INPUT "... Are you sure? (Y / N) "; A$
800 IF A$="Y" OR A$="y" THEN ELSE CLS:GOTO 720
810 '
820 CONSOLE 0,25,0,1 : LOCATE 0,0 : COLOR 6
830 PRINT USING" Program : Transfil.Exe(Bas)---DATA from &          & to : &
& : Exam: #";CODE$;CODE2$;EXAM : PRINT
840 CONSOLE 2,23,,1 : COLOR 7 :CLS
850 GOSUB *FILE.OP
860 FOR J=1 TO LOF(1)
870 GET #1,J
880 RC2=CVS(RC$):NO2#=CVD(NO$)
890 PRINT USING"####          #####";RC2;NO2#
900 FOR I=1 TO N
910 IF NO2#<>AT.NO#(I) THEN 960
920 ON EXAM GOSUB 1470,1560,1630          ... Selection fo E1$,E2$ or E3$
930 PUT #1,J
940 AT.NO#(I)=-100
950 GOTO 990
960 NEXT I
970 ON EXAM GOSUB 1100,1110,1120
980 PUT #1,J
990 NEXT J
1000 IF MID$(CODE2$,6,1)="B" THEN GOSUB *UNFINISH
1010 CLOSE #1
1020 PRINT "Would you stay on this program to put the data into Another File?"
1030 PRINT " ... If yes, hit 'Retrn' else hit any key." :IF INPUT$(1)=CHR$(13)
THEN 720
1040 '
1050 END
1060 '
1070 '
1080 '
1090 '
1100 RSET E1$=MK$(-1) : LSET CDR$=SPACE$(40): RETURN
1110 RSET E2$=MK$(-1) : LSET CDR$=SPACE$(40): RETURN
1120 RSET E3$=MK$(-1) : LSET CDR$=SPACE$(40): RETURN
1130 '
1140 '
1150 '
1160 '-----Random Access File -----
1170 '

```

```

1180 *FILE.OP
1190 OPEN "b:"+CODE2$ AS #1
1200 FIELD #1,4 AS RC$,8 AS NOS,12 AS SEI$,12 AS MEI$,2 AS CL1$,2 AS CL2$,4 AS E
1$ ,4 AS E2$,4 AS E3$,4 AS E4$,4 AS E5$,8 AS BG$,42 AS CDR$,4 AS SUM$,4 AS AVE$,4
AS ORD$,4 AS OPT$
1210 '
1220 '
1230 RETURN
1240 '
1250 *UNFINISH
1255 COLOR 5 : PRINT : PRINT
1260 FOR I=1 TO N
1270 IF AT.NO#(I)<0 THEN 1310
1280 PRINT AT.NO#(I)
1290 LPRINT AT.NO#(I),PNT(I),BG.N(I)
1300
1310 NEXT
1320 COLOR 7: RETURN
1330 '-----2nd Sequential File for January Test -----
1340 '
1350 OPEN "b:"+CODE$ FOR INPUT AS #1
1360 INPUT #1,N
1370 FOR I=1 TO N
1380 INPUT #1, RECRD,AT.NO#(I),CR.TR$(I),PNT(I),R.ANS
1390 PRINT USING "###";RECRD;
1400 PRINT " ";AT.NO#(I);CR.TR$(I);PNT(I)
1410 NEXT I
1420 CLOSE #1
1430 GOTO 700
1440 '----- Exam. Subroutine -----
1450 '
1460 '
1470 '
1480 PRINT BG.N(I);PNT(I);CR.TR$(I)
1490 RSET E1$=MK$$(PNT(I)):RSET BG$=MKD$(BG.N(I))
1500 LSET CDR$=CR.TR$(I)
1510 '
1520 RETURN
1530 '
1540 '
1550 '
1560 PRINT PNT(I);CR.TR$(I)
1570 RSET E2$=MK$$(PNT(I)):LSET CDR$=CR.TR$(I)
1580 RETURN
1590 '
1600 '
1610 '
1620 '
1630 PRINT USING"          ###      &
& ";PNT(I);CR.TR$(I)
1640 RSET E3$=MK$$(PNT(I)):LSET CDR$=CR.TR$(I)
1650 RETURN
1660 '
1670 '
1680 '

```

VI MON2

DATA ファイル (EXTENSION = "DAT") 任意の必要項目をコンソール・ディスプレイに表示させるためのプログラムである。

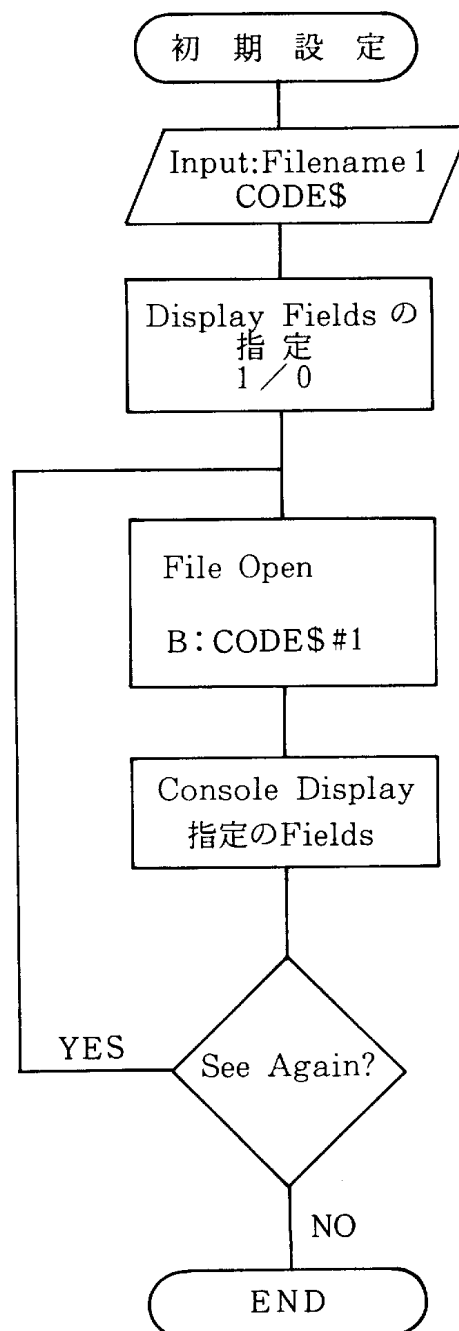
ファイル名 (CODE\$) を入力するに当っては、拡張子 "DAT" のファイル全を表示し、左上1個が反転している。この反転の位置はカーソル・キーで移動するから、必要とするファイルに、反転の位置を移動させ、リターン・キーを2回叩けば、ファイル名が入力できる (※ FILTAKE: 810—1400)。

項目を指定するためには、必要なものに数字1を入力する、必要がないものは単にリターン・キーを叩くだけでよい。訂正する時、最初の入力がディスプレイの上に消えないで残っているので、1の再入力が必要ないと思えるが、それは錯覚なので、改めて、必要項目には1を入力しなければならない。

データを印字するためには、このプログラムのフォーマットは必ずしも、適当でないので、そのためには、大量の件数を見易く配列し、プリンターに打ち出すための別のプログラムが必要である。それは、プリンター操作のためのエスケープ信号 (ESC Sequence) を各種利用しなければならないので、稿を改めて纏めるつもりである。

MON2 のフローチャートは次の通りである。これにはサブルーチン ※FILTAKE を省略した。

MON2



```

10 ' save"mon2",a
20 REM *****
30 REM   program MONITORING   v. 2.1   File Name "MONIT"
40 '
50 '
60 '
70 '
80 '
90 '
100 '
110 REM $file ,128
120 WIDTH 80,25 : CONSOLE 0,25,0,1,0
130 CLS :LOCATE 10,0 :COLOR 6
140 X$="Now Program NO 9: MONITOR ":X$=AKCNV$(X$):PRINT X$
150 COLOR 7:LOCATE 15,2
160 PRINT "Are you ready ?   Yes=Retrn   / No=Spc bar   ";

```

```

170 W$=INPUT$(1)
180 IF W$=CHR$(32) THEN END
190 IF W$=CHR$(13) THEN ELSE 170
200 COLOR 5 : LOCATE 5,3
210 GOSUB *FILTAKE :CONSOLE,,,1 :COLOR 7 :LOCATE 20,3
220 PRINT "You are reading Random File Data: ";CODE$
225 PRINT "You are sure? Yes=Return / No=Spc bar"
226 IF INPUT$(1)<>CHR$(13) THEN 210
230 CONSOLE 4,21 : CLS
235 LOCATE 10,5 : B=6
240 PRINT " Which are you going to review ?---Answer by Yes(1)/No(0)"
250 LOCATE 20,B+1:B=B+1:INPUT " NO " ;P1
260 LOCATE 20,B+1:B=B+1:INPUT " SE1 " ;P2
270 LOCATE 20,B+1:B=B+1:INPUT " ME1 " ;P3
280 LOCATE 20,B+1:B=B+1:INPUT " Class (1) " ;C1
290 LOCATE 20,B+1:B=B+1:INPUT " Class (2) " ;C2
300 LOCATE 20,B+1:B=B+1:INPUT " SUM " ;P4
310 LOCATE 20,B+1:B=B+1:INPUT " AVE " ;P5
320 LOCATE 20,B+1:B=B+1:INPUT " ORDER " ;P7
330 LOCATE 20,B+1:B=B+1:INPUT " BACKGROPUND " ;P8
340 LOCATE 20,B+1:B=B+1:INPUT " E1 " ;Q1
350 LOCATE 20,B+1:B=B+1:INPUT " E2 " ;Q2
360 LOCATE 20,B+1:B=B+1:INPUT " E3 " ;Q3
370 LOCATE 20,B+1:B=B+1:INPUT " E4 " ;Q4
380 LOCATE 20,B+1:B=B+1:INPUT " E5 " ;Q5
390 LOCATE 20,B+1:B=B+1:INPUT " CARD READER " ;Q6
400 : BEEP:BEEP
410 LOCATE 20,B+1:B=B+1:PRINT "*****": LOCATE 30,B
420 PRINT "Are you SURE? Yes=Return / No=other key ":IF INPUT$(1)<>CHR$(13) THEN
N 235
430 '
440 '
450 '----( Record length of 128 bytes for 126 in fact )-----
460 '
470 '
480 OPEN "B:"+CODE$ AS #1
490 FIELD #1,4 AS RC$,8 AS NO$,12 AS SE1$,12 AS ME1$,2 AS CL1$,2 AS CL2$,4 AS E1
$,4 AS E2$,4 AS E3$,4 AS E4$,4 AS E5$,8 AS BG$,42 AS CDR$,4 AS SUM$,4 AS AVE$,4A
S ORD$,4 AS OPT$
500 FOR I=1 TO LOF(1)
510 GET #1,1
520 NO#=CVD(NO$)
530 E1=CVS(E1$):E2=CVS(E2$):E3=CVS(E3$):E4=CVS(E4$):E5=CVS(E5$)
540 BG=CVD(BG$)
550 'SUM=CVS(SUM$):AVE=CVS(AVE$):ORD=CVS(ORD$):OPT=CVS(OPT$)
560 IF P1 THEN IF NO#>0 THEN PRINT USING "##### ";NO#;ELSE GOTO 730
570 IF P2 THEN PRINT USING "& & ";SE1$;
580 IF P3 THEN PRINT USING "& & ";ME1$;
590 IF C1 THEN PRINT CL1$;
600 IF C2 THEN PRINT CL2$;IF CL2$=CHR$(45)+ " " THEN T=T+1
610 IF P4 THEN PRINT USING "& & ";SUM$;ELSE PRINT " - ";
620 IF P5 THEN PRINT USING "& & ";AVE$;ELSE PRINT " - ";
630 IF P6 THEN IF TS>0 THEN PRINT USING "## ";TS;ELSE PRINT " - ";
640 PRINT USING "& & ";OPT$;
650 IF P7 THEN PRINT USING "& & ";ORD$;
660 IF Q1 THEN IF E1>0 THEN PRINT USING "### ";E1;ELSE PRINT " - ";
670 IF Q2 THEN IF E2>0 THEN PRINT USING "### ";E2;ELSE PRINT " - ";
680 IF Q3 THEN IF E3>0 THEN PRINT USING "### ";E3;ELSE PRINT " - ";
690 IF Q4 THEN IF E4>0 THEN PRINT USING "###. # ";E4;ELSE PRINT " - ";
700 IF Q5 THEN IF E5>0 THEN PRINT USING "###. # ";E5;ELSE PRINT " - ";
710 IF Q6 THEN IF CDR$<>"- " THEN PRINT CDR$; ELSE PRINT " - ";
720 IF P8 THEN IF BG<>0 THEN PRINT BG; ELSE PRINT " **";
730 PRINT
740 NEXT I
750 CLOSE #1
760 PRINT "Missing class2: ";T:T=0
770 INPUT "If you want to see it again, input ...Yes( 1 ) or No( 0 )";G:IF G=1
GOTO 480
780 '
790 END
800 '
810 *FILTAKE
820 CONSOLE 0,25,0,0,1
830 H1=0 : V1=6
840 LOCATE 0,5
850 FILES "b:*.dat"
860 *DISP.COL
870 COLOR(0, 0)-(77, 24),0
880 H2=H1+13
890 COLOR(H1, V1)-(H2, V1),4
900 LOCATE H2,V1,0
910 A$=INPUT$(1)
920 IF A$=CHR$(13) THEN *FIL.INPT
930 IF A$=CHR$(30) THEN GOSUB *UPWARD
940 IF A$=CHR$(31) THEN GOSUB *DOWNWRD
950 IF A$=CHR$(29) THEN GOSUB *LEFTWRD
960 IF A$=CHR$(28) THEN GOSUB *RIGHTWRD
970 '

```



```

980 GOTO *DISP.COL
990     RETURN          REM back to Prog. Monitor
1000 '
1020 *UPWARD
1030 V1=V1-1 : IF V1<6 THEN V1=6
1040 RETURN
1050 '
1060 *DOWNWRD
1070 V1=V1+1
1080 IF V1>24 THEN V1=24
1090 RETURN
1100 '
1110 *RIGHTWRD
1120 H1=H1+40
1130 IF H1>40 THEN H1=40
1140 RETURN
1150 '
1160 *LEFTWRD
1170 H1=H1-40
1180 IF H1<0 THEN H1=0
1190 RETURN
1200 '
1210 *FIL.INPT
1220 CODE$=""
1230     LOCATE H1,V1,1
1240     LINE INPUT A$
1250 FIL$=LEFT$(A$,12)
1260 '
1270 FF$=""
1280 FOR I=1 TO 9
1290     F$=MID$(FIL$,I,1)
1300 IF F$<>" " THEN FF$=FF$+F$ ELSE 1320
1310 NEXT
1320 IF FF$="" THEN *WARNING
1330 FF$=FF$+"."
1340 FOR I=10 TO 12
1350     F$=MID$(FIL$,I,1)
1360     IF F$<>" " THEN FF$=FF$+F$
1370 NEXT
1380     CODE$=FF$
1390 '
1400 GOTO 990
1410 '
1420 *WARNING
1430 LOCATE 10,0
1440 PRINT "There is No filename. Try again."
1450 GOTO 830
1460 '

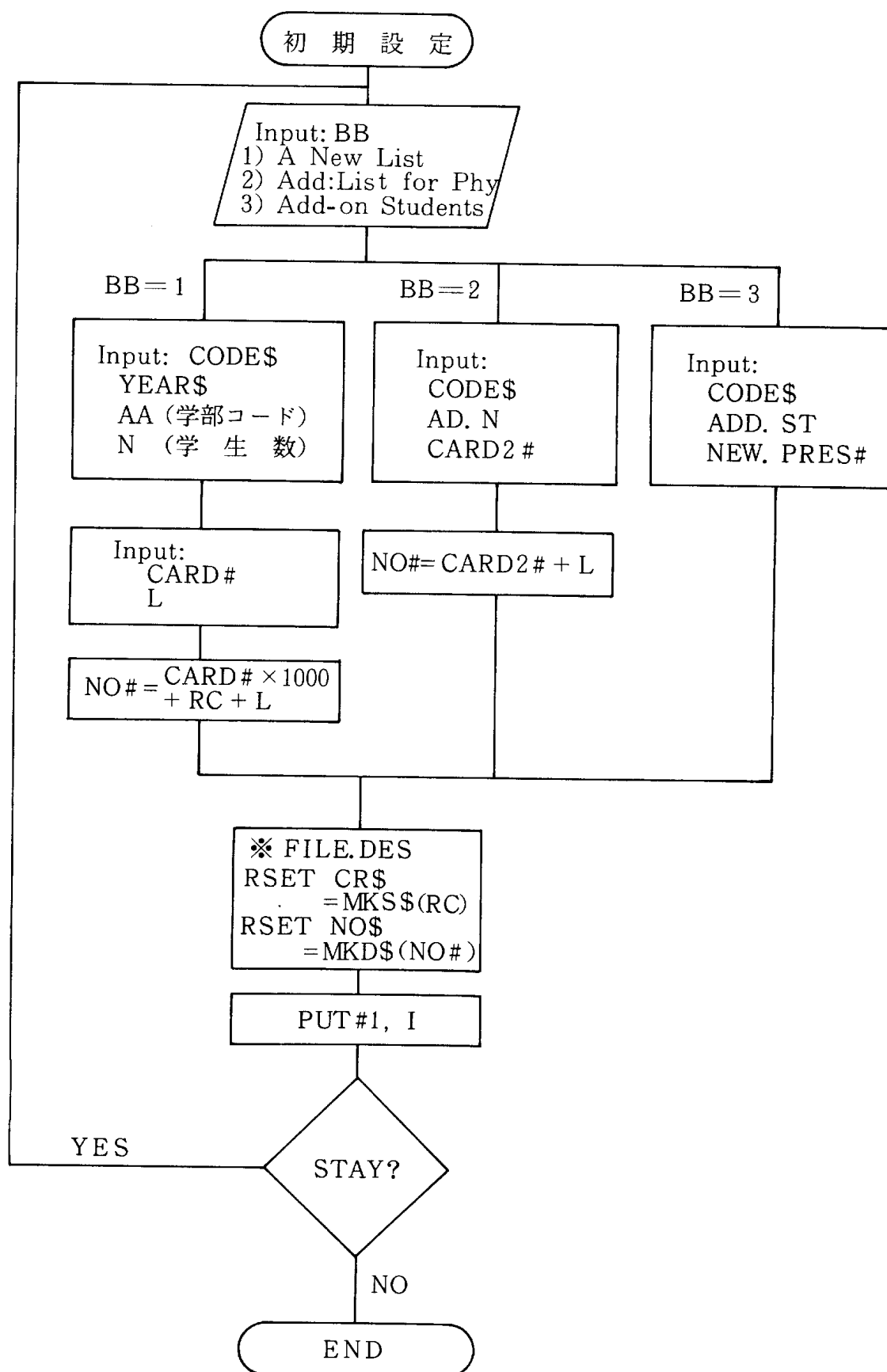
```

VII NUMBR2

TRANSFIL の項で述べた FIELD 配分を持つデータ・ファイルを作ると同時に、学籍番号を入力するプログラムである。文学部の学生は、われわれの資料対象ではない。その他学部は二分割され、それぞれ A, B のグループに分けられる。その学部の状況に応じた個々データ・ベースを作るとを目的としているので、1) A New List, 2) Add (itional) List for Physical Education の二種が基本的な方法として必要である。3) は復学、留年などによる既存のファイルへ追加する学生のレコードを作成するためのプログラムである。

フローチャートを示す。

NUMBR2



```

100 'save"numbr2",a
110 REM *****
120 '      INPUT OF      Student NUMBER
130 '      File name "NUMBR.BAS" V.3.0
140 '
150 '      List of Variables
160 '      YEARS$      :The Year of Anno Domini,ex. 1985
170 '      Y$          :2 End Bytes of Years$
180 '      CODE$       :Means the Department of Students to be numbered
190 '
200 '
210 REM $file ,128
220 CLS:CONSOLE 0,25,1,1:COLOR 6
230 PRINT TAB(15);"*****"
240 PRINT TAB(15);"***"
250 PRINT TAB(15);"***      INPUT OF SHUSSEKI No.      ***"
260 PRINT TAB(15);"***"
270 PRINT TAB(15);"*****"
280 FOR Z=1 TO 3:PRINT:NEXT Z :COLOR 7
290 INPUT"      Ready for NUMBERING... Yes(1)/ No(0)";RDY
300 IF RDY<>1 THEN 1430
310 *AGAIN
320 COLOR 6
330 PRINT "Which do you want to make? ..."
340 COLOR 7
350 PRINT " 1) a New List          2) Addit.List for Phy.Dept   3) Add-on Students
"
360 INPUT "... the number";BB
370 ON BB GOTO 400, 920, 1190      ' ... In case of BB=2, or BB=3, File of List
(1) supposed already existing.
380 '
390 COLOR 6
400 INPUT "      ... Input the Year of Entrance,as 1985.";YEARS$
410 COLOR 7
420 Y$=RIGHT$(YEARS$,2)
430 PRINT TAB(20);"YEAR : ";YEARS$;
440 PRINT TAB(40);" Y$ : ";Y$;
450 PRINT "Hit 'Retrn',if you are sure."
460 PRINT TAB(25):LINE INPUT A$ :CLS
470 PRINT:PRINT :COLOR 6
480 PRINT "Which Dept are you going to make the file of?"
490 PRINT:PRINT :COLOR 7
500 PRINT TAB(5);"1)Japanese C. Lit.Dept          2)English C. Lit.Dept."
510 PRINT TAB(5);"3)Psychology C. Lit. Dept.      4) Dept of Law"
520 PRINT TAB(5);"5) Commerce C.,Com. Dept        6)Management C.,Com. Dept"
530 PRINT TAB(5);"7) Physical Ed. Dept            8)Sociology Dept"
540 PRINT TAB(5);"9) Economic Dept              "
550 PRINT:PRINT :COLOR 4
560 INPUT ".....      Input the number";AA
570 COLOR 7
580 IF AA=1 THEN CODE$="JAP"
590 IF AA=2 THEN CODE$="ENG"
600 IF AA=3 THEN CODE$="PSY"
610 IF AA=4 THEN CODE$="LAW"
620 IF AA=5 THEN CODE$="COM"
630 IF AA=6 THEN CODE$="MAN"
640 IF AA=7 THEN CODE$="PHY"
650 IF AA=8 THEN CODE$="SOC"
660 IF AA=9 THEN CODE$="ECO"
670 PRINT "Would you have the dept Halved? If yes, hit 'Return'."
680 IF INPUT$(1)=CHR$(13) THEN ELSE 700
690 INPUT "Which 'A' or 'B' ";AFFIX$
700 CODE$=CODE$+Y$+AFFIX$+".DAT"
710 KEY 7,CODE$
720 COLOR 6
730 INPUT"How Many Students to Deal with";N
740 COLOR 7
750 INPUT" CARDINAL of Student Number (CARD*1000=NO)";CARD#
760 PRINT "      Input the base number( ex. 0 or 272) ";:INPUT L
770 PRINT"-----"
780 CLS
790 LOCATE 0,0 :COLOR 5
800 PRINT USING"DEPT. CODE:8      & NO.to Start from*: ####### ,NUMBER of s
tudents: ####";CODE$;CARD#*1000+L+1;N
810 INPUT "..... (Y/N)";AA$
820 IF AA$="Y" OR AA$="y" THEN ELSE 470
830 GOSUB *FILE.OP
840 CONSOLE 2,23,0,1
850 FOR I=1 TO N
860 RC=1:NO#=CARD#*1000+RC+L
870 PRINT RC,NO#
880 GOSUB *FILE.DES
890 NEXT I
900 CLOSE #1

```

```

910 GOTO 1420
920 '-----Additional List(Phy. Dept) File Making-----
930 IF CODE$="" THEN ELSE 980
940 COLOR 4
950 FILES "b:*.dat"
960 COLOR 7
970 INPUT "... CODENAME FOR NO# to be added to";CODE$
980 PRINT CODE$
990 PRINT "... Sure ? Hit Retrn Key, if OK." :A$=INPUT$(1)
1000 IF A$=CHR$(13) THEN ELSE 940
1010 COLOR 6
1020 INPUT "How many students to add ";AD.N
1030 COLOR 7
1040 INPUT "Cardinal II of students to add (ex. 9188500)";CARD2#
1050 PRINT "Dept. Code: ";CODE$;" Cardinal: ";CARD2#;" Number of Students: ";A
D.N
1060 GOSUB *FILE.OP
1070 L=0
1080 FOR I=LOF(1)+1 TO LOF(1)+AD.N
1090 L=L+1
1100 RC=I : NO#=CARD2#+L
1110 PRINT RC,NO#
1120 '
1130 '
1140 GOSUB *FILE.DES
1150 '
1160 NEXT I
1170 CLOSE #1
1180 '
1190 '----- Add-on Students -----
1200 INPUT " Input CODE$ if addendum is Irregular NO#.";CODE$
1210 IF CODE$<>"" THEN ELSE 1410
1220 PRINT CODE$
1230 PRINT " ....Sure, if yes, hit 'Return'.";: IF INPUT$(1)=CHR$(13) T
HEN ELSE 1200
1240 IF CODE$<>"" THEN ELSE 1410
1250 PRINT
1260 INPUT "How many students to be added";ADD.ST
1270 GOSUB *FILE.OP
1280 FOR I=LOF(1)+1 TO LOF(1)+ADD.ST
1290 '
1300 '
1310 '
1320 INPUT " New Pres. No. ....";NEW.PRES#
1330 INPUT " ....Sure? (Y)/(N)";AA$
1340 IF AA$="Y" OR AA$="y" THEN ELSE 1320
1350 PRINT USING"New Rec.No. #### Pres. No. #####" ;I;NEW.PRES#
1360 RC=I: NO#=NEW.PRES#
1370 GOSUB *FILE.DES
1380 CONSOLE 15,25,1,1 : CLS: LOCATE 0,16
1390 NEXT I
1400 CLOSE #1
1410 CONSOLE 0,25,1,1
1420 CLS:PRINT:PRINT: LOCATE 0,10 : COLOR 6
1430 PRINT"Are you to stay on this program to add students to the same file ?"
1440 COLOR 7
1450 INPUT " .... (Y)/(N)";AA$
1460 IF AA$="Y" OR AA$="y" THEN GOTO *AGAIN
1470 '
1480 END
1490 '
1500 '-----Random Access Subroutine-----
1510 '----- ( Record length of 128 bytes for 126 in fact )-----
1520 '
1530 '
1540 *FILE.OP
1550 OPEN "b:"+CODE$ AS #1
1560 FIELD #1,4 AS RC$,8 AS NO$,12 AS SEI$,12 AS MEI$,2 AS CL1$,2 AS CL2$,4 AS E
1$,4 AS E2$,4 AS E3$,4 AS E4$,4 AS E5$,8 AS BG$,42 AS CDR$,4 AS SUM$,4 AS AVE$,4
AS ORD$,4 AS OPT$
1570 RETURN
1580 '
1590 '
1600 '
1610 *FILE.DES
1620 RSET RC$=MK$(RC):RSET NO$=MK$(NO#)
1630 PUT #1,I
1640 RETURN

```

VIII BATCH FILE

MS-DOS の BATCH FILE が有効な機能であることは、今更、言うまでもない。I 章の MS-DOS の項で述べたような、チャイルド・プロセスに替わるバッチ・ファイルなら、次のような簡単な操作で作ることができる。

```
COPY CON BASIC. BAT ^\
```

```
N88BASIC / F: 4 /S: 128 ^Z
```

^Z は EOF であり、F. 10 キーを叩けば入力できる。これだけで BASIC という Filename のバッチ・ファイルが作成される。その中身は [N88 BASIC /F: 4 /S: 128] で、[BASIC] とプロンプトに応じて入力すれば、それが実行されることになる。

比較的長いものでもライン・エディタ (EDLIN. COM) を使って書くことができるが、ワープロ (Word Processor) や高速エディタを利用する方が便利である。但し、バッチ・ファイルでカラーを利用したり、特殊なエスケープ・シーケンスを入力するには、高速エディターが最も有効であるのは、言うまでもない。

4 月の統一テスト、後期の共通テストそれぞれに必要なプログラム群が確定しているので、それを一括し、順序に序じてロードし、実行するとすると、一枚のシステム・ディスクに起動と同時に、一連のプログラムを 1 回の選択で走らせることが可能である。

選択のスイッチ・ボードは MENU. TXT として次の様に作ることができる。

- 1) Spring Programs
- 2) Winter Programs
- 3) Classification Programs
- 4) Graphic Programs

Input the number of processing you want to set in.

次に AUTOEXEC. BAT を次の様に作っておくと、20 行目の 50 桁のと

ころにカーソルが現れ、入力を促す。

```
ECHO OFF
```

```
CLS
```

```
PROMPT $A
```

```
TYPE MENU. TXT
```

```
PROMPT $e [20; 50H
```

TYPE コマンドでコンソールの上に示される MENU. TXT がスイッチ・ボードであるかの様な錯覚を与えるにすぎないのだが、`1' の入力に対応する 1. BAT というバッチファイル、`2' に対応する 2. BAT というファイルが存在するので、プログラムは進行する。

1. BAT は次の如く、

```
unittest
```

```
prompt $n$g
```

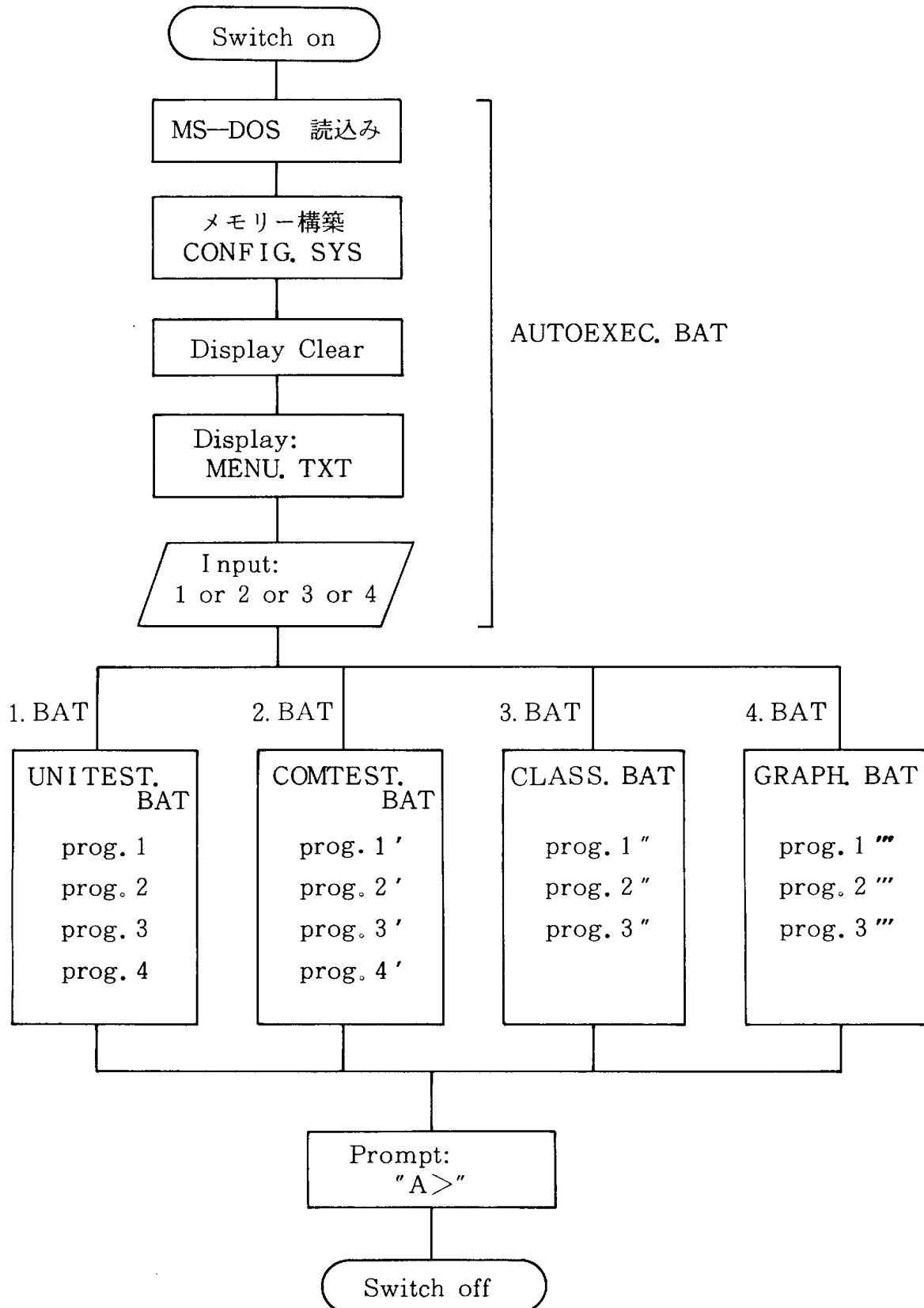
2. BAT も次の如く、

```
comtest
```

```
prompt $n$g
```

簡単でよい。両者の最後のコマンドはプロンプト“A>”を出力するためのもので、実際のプログラムの制御は UNITEST. BAT, COMTEST. BAT に担わせるのである。選択の分岐を図で示せば次の如くである。

BATCH FILE



IX 将来構想と文献

この報告では、業者が提供する EBCDIC コードの読み取っただけの未処理データを解析、採点して、データ・ベースに収納するのに必要なプログラムを中心に扱った。業者 C. S. 社が納品するのは、テストを実施した日から、中二日置いて、三日目の午前中であるが、パソコン一台（カード・リーダ付）で実施しても、われわれのプログラムでは、受験者数 3,000 人程度を対象として、約 3 時間で終了する。但し、これには、入学者報告に基づく、いわゆるクロス・レファレンスの処理は計算に入っていない。現在利用されている習熟度別クラス編成に必要なデータ処理に限って言っているのである。

パソコンの配列エリアは 16 ビットでは 64 K に限定され、この空間を MS-DOS 及びプログラムがまず占居するから、無闇に、長大なプログラムをロードすると、データを格納するエリアが不足することになる。従って、仕事を適切な大きさに分割し、その段階で、プログラムを取り替えながら、作業を進めることが要求される。プログラムの取り替えに要する時間を短縮するために、RAM ディスクを実際は活用しているのだが、それについての記述はここでは割愛した。

また、N88 DISK-BASIC (86) (MS-DOS 版) の Vr. 4 は P コード、インタープリンター方式でありながら十分高速である。とは言え、不自由はないが、プログラム作成はかなり面倒でもある。それが、入学者報告とのクロスレファレンスへの挑戦を妨げていることも否めない。

しかし最近コンピュータ演習室が備えた dBASE III PLUS を少し扱ってみて、そのプログラミングの簡略さと精緻さは、このテスト処理に対しても、十分機能し得るのではないと思われる。コマンドが豊富で、処理そのものは勿論、プリンター出力も、必要なフォーマットが作れそうである。それ自体、多目的なデータ・ベースとして既に定評があり、コンピュータ言語の学習は避けたいと言いながらも、データ・ベースで文献、その他の研究資料を活用したいという人はこれから増えてくる情勢であるから、テスト処理に最大限に dBASE III PLUS を活用することは、大いに、意義が

あると思える。

dBASE III PLUS の長所を思いつくままに列記すると、

- 1) 配列宣言をしなくてよい。
- 2) Data 構造は自由にしかも短時間で改造できる。
- 3) BASIC の文法と根本はあまり変らない。
- 4) 扱えるレコード件数が膨大で、多量である。
- 5) そのデータを他の既製ソフトを使って、再処理、あるいは、グラフィックなどの再表現が可能でもある。

の如くで、周辺機、例えばハード・ディスクが揃えば、パソコンによるデータ処理は、更に簡単にかつ、時間も短くなって行くと期待される。

ともあれ、業者処理を再処理するという煩わしさは本年度で終る。というのは、統一テストは来年度も続けられるが、その際、カード・リーダ数台を使って、まず、正確にカードを読み取って採点し、その後に、多項目の分析を業者に依頼することになったからである。来年度においては、学籍番号、回答、得点、所属クラス名を項目とする、学部各の電子データ (Floppy Disk) を業者が読み取って処理する手順を踏む。その結果英語クラス名簿は、本年度より、一週間近く早く実現するのではあるまいか。

最後に利用した文献を挙げておく。

「MS-DOS 2.0 ユーザーズ・マニュアル」日本電気 (1985)

アスキー出版局版「標準 MS-DOS ハンド・ブック」アスキー (1985)

アスキー書籍編集部「MS-DOS 3.1 ハンド・ブック」アスキー (1986)

De Voney, Chris: *The MS-DOS User's Guide*, Que Corporation, (Indianapolis, 1984)

King, Richard A.: *The MS-DOS Handbook*, SYNBEX, (Berkely, 1985)

「よこぐま vr. 2.01 操作説明書」COMPUS

「N88-日本語 BASIC (86) (MS-DOS 版) コンパイラ、ユーザーズ・マニュアル」
日本電気

「N88-日本語 BASIC (86) (MS-DOS 版) ユーザーズ・マニュアル」日本電気

ピート・メテヴェーレス 「MS-DOS ファイル整理学」アスキー出版局 (東京, 1986)

ピート・メテヴェーレス 「MS-DOS バッチ・プログラム集 the Pemet Utili-

ties」技術評論 (東京, 1987)

大澤昭二・春田正夫「dBASE III PLUS ハンディ・マニュアル」新星出版 (東京, 1986)

村山集治「dBASB III PLUS 例題方式練習テキスト」エーアイ出版 (東京, 1988)

井上寿雄「マイコン OA 用語辞典」電波新聞 (東京, 1984)

Dowing, Douglas: *Encyclopedia of Computer Terms*, Barron's, (Woodbury, 1983)

S. Curren & R. Curnow: *The Penguin Computing Book—A Complete and Comprehensive Guide to Computing*, Penguin Books, (Harmondsworth, 1983)

Data Processing Programs written in N88 BASIC(86) under MS-DOS

Toshihiro Namba

Synopsis

The present article is about BASIC programs with auxiliary tool softs under MS-DOS vr. 2. 11, a Disk Operating System adapted and presented by NEC for their PC 98 Model. We have composed these programs to try to mark the answers from the first grade students, fresh from Highschools, which are offered in form of EBCDIC by a processing service company. With a view of making the data readable under MS-DOS, we have made use of YOKOGURUMA, a conversion software manufactured by COMPS. In the following chapters, what we have observed while the programs being run is briefly explained and commented with a simple kind of flowcharts.

- I. MS-DOS & BASIC.COM
- II. YOKOGURUMA, a Device of Conversion to ASCII
- III. FILREAD
- IV. TRETAPR
- V. TRANSFIL
- VI. MON2
- VII. NUMBR2
- VIII. BATCH FILE
- IX. Prospective Remarks as Conclusion & Bibliography